NASA-TM-112242

7N-34-TM
021677

NASA

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ARC 275 (Rev Feb 81)

A Virtual Environment for the Exploration of Three Dimensional Steady Flows

Steve Bryson[1] and Creon Levit[2]

Report RNR-90-013, October 1990

NAS Applied Research Laboratory
NASA Ames Research Center, Mail Stop T-045-1
Moffett Field, CA 94035

Abstract

We describe a recently completed implementation of a virtual environment for the analysis of three dimensional steady flowfields. The hardware consists of a boom-mounted, six degree-of-freedom head position sensitive stereo CRT system for display, a VPL dataglove(tm) for placement of tracer particles within the flow, and a Silicon Graphics 320 VGX workstation for computation and rendering. The flowfields that we visualize using the virtual environment are velocity vector fields defined on curvilinear meshes, and are the steady-state solutions to problems in computational fluid dynamics. The system is applicable to the visualization of other vector fields as well.

1 Introduction

Visualization of three dimensional flowfields is difficult. Even if we restrict our attention to steady three-dimensional physically realistic velocity fields, complicated

[1] Sterling Software, Inc.
[2] NASA Ames Research Center

geometrical and topological situations arise. For example, multiple vortices, recirculation bubbles, and chaotic flows within vortex breakdown have all been observed in computer simulations of steady flows.

One popular technique for visualizing flowfields[3] is by looking at streamlines. In this paper, we describe an implementation of a virtual environment for interactively visualizing three dimensional flowfields using streamlines. First, we introduce the concept of streamlines and some terminology. Then, we discuss the required functionality of our virtual environment. We describe the implementation of our environment - first the hardware and then the software. We review the performance of our implementation, and, finally, we conclude with some comments on what we have learned and discuss our future plans.

2 Definitions and Requirements

A streamline is a curve that is everywhere tangent to the velocity field. It is an integral curve of that vector field. Given any point P in the flowfield where the velocity is not zero, one and only one streamline, S, passes through P. Part of the S extends from P in the forward direction, parallel to the local velocity, and part of S extends from P in the backward direction, antiparallel to the velocity. We call P the *seed-point* of the streamline, and we say that P *generates* S. In a steady flowfield, a streamline is the same as a particle path (or particle trace), and so we shall use the terms interchangeably.

While a single streamline may show comparatively little, several carefully placed placed streamlines may together shed considerable light on the dynamics of the flow. [see figure 1] But it is not obvious where to place the seed points that generate the most enlightening streamlines. It can differ radically from one flowfield to another.

There are several approaches to the placement of seed points. One approach is to have the computer automatically place them based on analysis of the flow field. For example, seed points may be placed near critical points of the vector field topology[1, 2], or near local maxima of interesting scalars such as helicity [3]. However, this approach to placement is still in its infancy. A second approach is simply to give the user flexible, rapid, interactive control over the placement of seed points[4]. Our virtual environment supports this second approach.

Besides rapid placement of seed points, our virtual environment allows for quick, intuitive repositioning and deletion of existing seed points. Multiple seed points can be grouped together into "rakes", and repositioning and deletion of these rakes is supported.

Since fluid dynamic phenomena occur over a large range of scales (several orders

[3] Unless otherwise noted, when we refer to a "flowfield", we mean a steady-state numerical solution to a three dimensional computational fluid dynamics (CFD) simulation, and in particular, the velocity vector field part of the solution

of magnitude in space) navigation through the virtual environment takes on new difficulties. In additional to "standard" head-position and head-orientation sensitive viewing capabilities, our virtual environment supports the ability to rapidly change one's scale, or the scale of the environment. Thus, a user of the environment can shrink herself so that she is completely surrounded by some small vortex, or enlarge herself so that the entire flowfield fits within her hand.

While multiple streamlines can give a good qualitative picture of the flow, streamlines are three dimensional curves that may wind through space in complex and even chaotic ways. To get a good mental picture of the flowfield from a set of streamlines, cues illuminating the streamlines' three dimensional geometry are important. Without these cues, ambiguity can result, leading to poor perception of the flowfield. [see figure 1]

In our virtual environment we use a combination of several techniques to disambiguate the three dimensional structures of streamlines. Real-time three dimensional rendering in response to head position and orientation provides perspective, motion parallax, and vestibular cues. Z-buffering enhances the realism of the image through selective obscuration. Wide field-of-view optics provide input to the peripheral visual field and give a realistic, compelling optical flow. Stereo display provides binocular parallax and further widens the field of view. Finally, feedback as to the position of the hand and animation of the streamlines themselves provide additional cues.

3 Hardware

Perhaps the most interesting hardware component of our virtual environment is the boom-mounted display [see figure 2]. This boom supports two small CRTs on a counterweighted yoke attached through six joints to a base. It is manufactured by FakeSpace Labs of Menlo Park CA., and fashioned after the prototype developed earlier by Sterling Software, Inc. for NASA Ames research center[5].

The boom is an alternative to the popular head-mounted LCD display systems that were pioneered at NASA Ames and are now in use throughout the country. The main advantage of the boom is that real CRTs can be used for display in spite of their mass, since none of the weight of the displays is born by the user. CRTs have much better brightness, contrast, and resolution than standard liquid crystal displays.

The CRTs are mounted on the "head" of the boom, along with the wide field optics and two repositionable handles. Six-degrees of freedom of motion are provided by the the gimbals and joints of the boom, in a smooth and force-free manner. Within a very wide range, the user can continuously change the three dimensional position and orientation of the head of the boom. The position and orientation information is based on the the current state of the six joints angles. These angles are sensed by optical encoders at the joints and fed into a microprocessor in the base of the boom, which formats the information and send it out an RS232 port. No magnetic field emitters or sensors are used, and hence the boom information is precise, repeatable, and insensitive to the electromagnetic environment. Calibration is trivial.

Currently, the monitors on the boom are monochrome. The boom accepts two RS170 signals, one for the left eye and one for the right.

The motion of the boom is relatively effortless and completely smooth. First-time users are universally surprised that a structure this size moves so easily. Admittedly, compared with head-mounted displays, the footprint of the device is large and the freedom of motion restricted. But when used from a sitting position in a wheeled office chair, it provides ample freedom of motion. With the user standing it is quite usable as well. There are no straps, no weight on the head, and it takes about one second to disengage from the device and hand it to another user.

In addition to the boom head position and orientation, the user's hand position, orientation, and finger joint angles are sensed using a VPL dataglove model II (tm), which incorporates a Polhemus 3Space (tm) tracker. The finger joint angles are combined and interpreted as gestures by a low level of the software. The glove requires recalibration for each user, and the polhemous tracker on the glove is, unfortunately, sensitive to the room's electromagnetic environment. Nevertheless, this part of the system works reliably and satisfactorily once calibrated for a users hand and a room's magnetic peculiarities.

The keyboard and mouse are also used as input devices to the virtual environment. The boom can be easily swung away from the user's eyes and her attention refocussed on the normal computer screen. The user, who is seated, can then return to typing and interacting with the computer in the usual way. For small amounts of typing, and for controlling the mouse, the glove need not be removed, since it is quite thin and flexible.

The computational and rendering power for our virtual environment is provided by a Silicon Graphics Iris 320 VGX system. This is a dual processor system with two 33 MHz RISC processors (MIPS R3000 CPUs with R3010 floating point chips). The performance of the machine is rated at approximately 50 Million instructions per second (50 MIPS) and 8.7 Million floating point operations per second (8.7 64-bit linpack MFLOPS). Our system has 48 MBytes of memory.

The VGX also has parallel hardware rendering pipelines. The rated graphics performance of our system is around one million 3D triangles transformed, clipped, projected, lit with multiple lights, Phong-shaded, and displayed per second. The system has over 200 bits per pixel of frame buffer memory. We make use of only 48 bits per pixel - eight bits of red and eight bits of green in each of two buffers (double buffering), and 24 bits of Z-buffer.

Stereo display on the boom is handled by rendering the left eye image using only shades of pure red (of which 256 are available) and the right eye image using only shades of pure green. When the green (second, right-eye) image is drawn, it is drawn using a "writemask" that protects the bits of the red image. The Z-buffer bit planes are cleared between the rendering of the left- and right-eye images, but the color (red) bit planes are not cleared. Thus, the end result is separately Z-buffered left- and right-eye images, in red and green respectively, on the screen at the same time with the appropriate mixture of red and green where the images overlap.

The 1024x1280 pixel RGB video output of the VGX is converted into RS170 component video in real time using a scan converter[4]. The red RS170 component is fed into the left eye of the boom, and the green RS170 component into the right eye. The synch is fed to both eyes. Since the boom CRTs are monochrome, we see correctly matched (stereo) images.

4 Software

The program for the visualization of streamlines using the virtual environment is based on code written by Jeff Hultquist[4]. His code has been modified in several ways: The graphics are rendered in stereo from a point of view determined by the boom. The interface to the streamlines is based on the glove position and gesture. Finally, there is an interface allowing the flow data to be moved relative to the user.

The position and orientation of the CRTs on the boom head are determined by optical encoders mounted on the six boom joints. The output of each encoder is linearly related to its respective joint angle. These six joint angles are read by the host computer system and are converted into a standard 4x4 position and orientation matrix. This conversion is the result of six successive translations and rotations. The rotations are rotations about the local axis of the corresponding joint by the angle read at that joint, and the translations are by the distances between joints. In the position and orientation matrix, the position is measured in meters from the base of the boom. Once the position and orientation matrix of the boom is constructed, that matrix is used to render the graphics.

The graphics is rendered in stereo from the point of view of the boom's viewers by inverting the boom's position and orientation matrix, translating to the left or right by half the distance between the eyes (depending on which eye's view is being drawn), and concatenating with a precomputed perspective matrix. The resulting matrix is put on the graphics transformation stack before the rendering of graphics for each eye. Thus the entire view must be rendered twice.

The alignment of the resulting images in the viewer must be correct for attaining a proper stereo effect. This is currently accomplished by defining separate viewports for the rendering of the graphics for each eye, with the horizontal position of each viewport controlled by a variable. To determine the proper alignment for stereo perception by an individual, two parallel vertical lines, one in red for one eye and the other in green for the other eye, are drawn in the same position relative to the

[4] While the VGX can put out RS170 directly by setting a software switch, we have found that scan converting the higher resolution 1024x1280 image using dedicated hardware provides spatial and temporal antialiasing, and consequently noticeably better image quality when viewing with the boom

rendering viewport for each eye. As these lines are in the same position relative to the viewports, they should appear to the user as a single line at infinity when the viewports are properly aligned. The viewports are moved horizontally under user control until these two lines are perceived by the user as a single line infinitely far away.

Before rendering the graphics data, another transformation embodying a rotation and translation is concatenated onto the transformation stack. This allows the data to be in an arbitrary position and orientation with respect to the coordinate system of the boom. This transformation can be determined in a variety of ways, in particular it enables the manipulation of the entire graphics data via the VPL dataglove. This transformation is called the data coordinate transformation.

The primary use of the VPL dataglove is in the placement of the seed points of streamlines. The interface is gesturally based. When the glove position is matched with a streamline seed position and the glove gesture is that of a fist, the seed point is 'picked up' and follows the position of the glove until the fist gesture is released. Throughout this time the streamline for this seed point is recomputed and rendered, allowing the user to observe the streamline as it is moved from place to place. When another gesture is performed, currently pointing with the index and middle finger simultaneously, a new streamline seed point is placed at the current position of the glove. This interface has also been generalized to groups of streamlines.

The above interface actually spans three coordinate systems: the glove position is in boom coordinates, the graphics data is in data coordinates, and the streamlines are in computational coordinates. To transform from boom coordinates to data coordinates, the glove position vector is multiplied by the inverse of the data coordinate transformation. The resulting vector is in data (x, y, z) coordinates.

The streamlines are in computational (ξ, η, ζ) coordinates which are defined on a discrete curvilinear computational grid. The (x, y, z) values in data coordinates for the grid (ξ, η, ζ) vertices are stored in a three-dimensional array. The (ξ, η, ζ) values for the glove's (x, y, z) coordinates in data coordinates are determined using a table search and then refined by inverse interpolation. First the nearest grid vertex to the given (x, y, z) is determined through a simple two step search of the vertex array. This gives a (ξ, η, ζ) value for that vertex. A first order inverse interpolation is then performed using the (x, y, z) values of the neighboring (ξ, η, ζ) points to determine the (ξ, η, ζ) values of the glove position. We are aware that this technique fails on grids that are highly curved and stretched, and we are presently implementing a more general method.

Once the seed point's computational space coordinates are known, these coordinates are used as an initial condition for generating a streamline. The streamline is generated by integrating a system of three ordinary differential equations:

$$\begin{aligned}\dot{\xi} &= u(\xi, \eta, \zeta)\\ \dot{\eta} &= v(\xi, \eta, \zeta)\\ \dot{\zeta} &= w(\xi, \eta, \zeta)\end{aligned}$$

where $u(\xi,\eta,\zeta)$, $v(\xi,\eta,\zeta)$, and $w(\xi,\eta,\zeta)$, are the components of the velocity vector, in computational coordinates, at the point (ξ,η,ζ). The velocity within a grid cell is obtained by trilinear interpolation of the velocities at the eight vertices of the cell. Integration is carried out using an explicit modified Euler method with the step size adjusted to take approximately five steps per grid cell. Integration is continued until the streamline either leaves the boundaries of the grid, or a fixed number of steps have been exhausted.

The computational coordinates of each point (ξ_i,η_i,ζ_i) along a streamline are converted back their physical coordinate values (x_i,y_i,z_i) and then displayed, connected by lines. This method is fast enough to drag streamlines through the field interactively, and inaccurate only near extreme velocity or metric discontinuities.

5 Performance

The current system performs with a stereo frame rate of approximately 15 frames/second drawing two grid planes and as many as ten tracers. All graphics is rendered with lines. When drawing only one eye, the frame rate increases to about 25 frames/second, and so the primary bottleneck is the double rendering of two grid planes. The system is thus graphics bound when there is no streamline manipulation. When manipulating a streamline, necessitating the computation of a full streamline in each frame, the frame rate drops slightly. When a rake of five streamline is being manipulated, necessitating the computation of five full streamlines in each frame, the frame rate drops to about 8 frames/second. The computation of streamlines will be significantly sped up by the use of the multiple processor capacity of the Iris 320 system. Currently only one processor is used.

6 Future Plans

Our future plans involve hardware upgrades to the current system and the extension of the system to the visualization of unsteady flows.

We plan to upgrade the boom to contain color displays of the highest feasible resolution, pending a safety study on the effects of having color monitors in such close proximity to the eye. Wider field optics will be used on the color boom, and we may then be forced to confront the issue of rectifying the barrel distortion at the edges of the display. We will also use a standard head-mounted color LCD display in the same virtual environment for comparison with the boom.

The most exciting future prospect is the extension of the system to the visualization of unsteady flows[6]. Many additional visualization techniques will then be applicable.

Though the computational requirements, memory requirements, and graphics requirements for visualizing 3D unsteady flowfields far exceed those for visualizing

3D steady flowfields, we believe these requirements can be met, at least for some smaller datasets, by a maximally configured Silicon Graphics VGX machine (with eight CPUs, and 256 MBytes memory). For larger flowfields, or nonperiodic flowfields where hundreds of separate timesteps are required, we may need to use a supercomputer to store the field and advect the particles. We will then send the updated three dimensional particle coordinates to the IRIS over ULTRAnet, where they will be transformed by the user's head position and displayed.

7 Summary

We have described our implementation of a virtual environment for visualizing three dimensional steady flows using streamlines. A special hardware component of our environment is a boom-mounted six degree-of-freedom head-position sensitive stereo display. Computation and rendering is handled by a Silicon Graphics 320 VGX computer system. Initial positions in space for streamlines are specified by the hand position in the environment using a VPL dataglove. We find system performance adequate for investigating the flowfield geometry from the "inside".

References

[1] Jim Hellman and Lambertus Hesselink. Surface representations of two- and three-dimensional fluid flow topology. To appear in *Visualization '90 Conf. Proc.*, 1990.

[2] Al Globus and Creon Levit. Computation and display of the topology of three-dimensional vector fields. To appear in *Proc. NASA Computational Fluid Dynamics Conference*, 1991.

[3] Yuval Levy and Arnan Seginer. Graphical representation of three-dimensional vortical flows by means of helicity density and normalized. In *AIAA 6th Applied Aerodynamics Conference*, 1988. Paper no. AIAA-88-2598.

[4] Jeff Hultquist. Interactive numerical flow visualization using streamsurfaces. To appear in Ahmed K. Noor, editor, *Symposium on Computational technology for Flight Vehicles*, November 1990.

[5] I.E. McDowall, M. Bolas, S. Pieper, S.S. Fisher, and J. Humphries. Implementation and integration of a counterbalanced crt-based stereoscopic display for interactive viewpoint control in virtual environment applications. In John Merrit and Scott Fisher, editors, *Proc. SPIE Conf. on Stereoscopic Displays and Applications*, 1990.

[6] Dennis Jespersen and Creon Levit. Numerical Simulation of flow past a tapered cylinder. to appear in *AIAA 29th Aerosciences Meeting*, 1991.